Boolean array operations and piecewise functions

Software 1 – Python Exercises for Mathematics

Autumn 2023

Sakari Lukkarinen

Metropolia University of Applied Sciences

Metropolia

Contents

- Examples of boolean operations
  - x > 5
  - f(x) > 2
  - More examples
- Graphing boolean values
  - Multiplying numerical array with boolean values
  - Piecewise continuous function
- Next steps

Example: x > 5

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
x = np.arange(10)
b = x > 5
print('x = ', x)
print('b = ', b)
```

```
x =  [0 1 2 3 4 5 6 7 8 9]
b =  [False False False False False False  True  True  True  True]
```

Example: y > 2

[3]:
```
x = np.arange(-5, 5)
y = 0.25*x**2
b = y > 2
print('y = ', y)
print('b = ', b)
```

```
y =  [6.25 4.   2.25 1.   0.25 0.   0.25 1.   2.25 4.  ]
b =  [ True  True  True False False False False False  True  True]
```

More examples

[7]:
```
# Create an example array
x = np.arange(10)
x
```

[7]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

[8]:
```
# Which elements are less than 4?
1.0*(x < 4)
```

[8]:
```
array([1., 1., 1., 1., 0., 0., 0., 0., 0., 0.])
```

[9]:
```
# Calculate x**2, when x < 4
x**2*(x < 4)
```

[9]:
```
array([0, 1, 4, 9, 0, 0, 0, 0, 0, 0], dtype=int32)
```

[10]:
```
# Calculate:
# x, when x < 4
# x**2, when x >= 4
x*(x < 4) + x**2*(x >= 4)
```

[10]:
```
array([ 0,  1,  2,  3, 16, 25, 36, 49, 64, 81])
```

step-function or boolean operations

[11]:
```
# Display x, when 3 < x < 8, otherwise 0
x*((3 < x) & (x < 8))
```

[11]:
```
array([0, 0, 0, 0, 4, 5, 6, 7, 0, 0])
```

[12]:
```
# Display 5, when 3 < x < 8, otherwise 0
((3 < x) & (x < 8))*5
```

[12]:
```
array([0, 0, 0, 0, 5, 5, 5, 5, 0, 0])
```

[13]:
```
# Define a step function
def step(x):
    return 1.0*(x>=0)
```

[14]:
```
# Calculate
# x/2, when 3 <= x < 6
# otherwise 0
(step(x-3) - step(x - 6))*x/2
```

[14]:
```
array([0. , 0. , 0. , 1.5, 2. , 2.5, 0. , 0. , 0. , 0. ])
```

[15]:
```
# Same results, but now calculated without step(x) function
((x >= 3) & (x < 6))*x/2
```

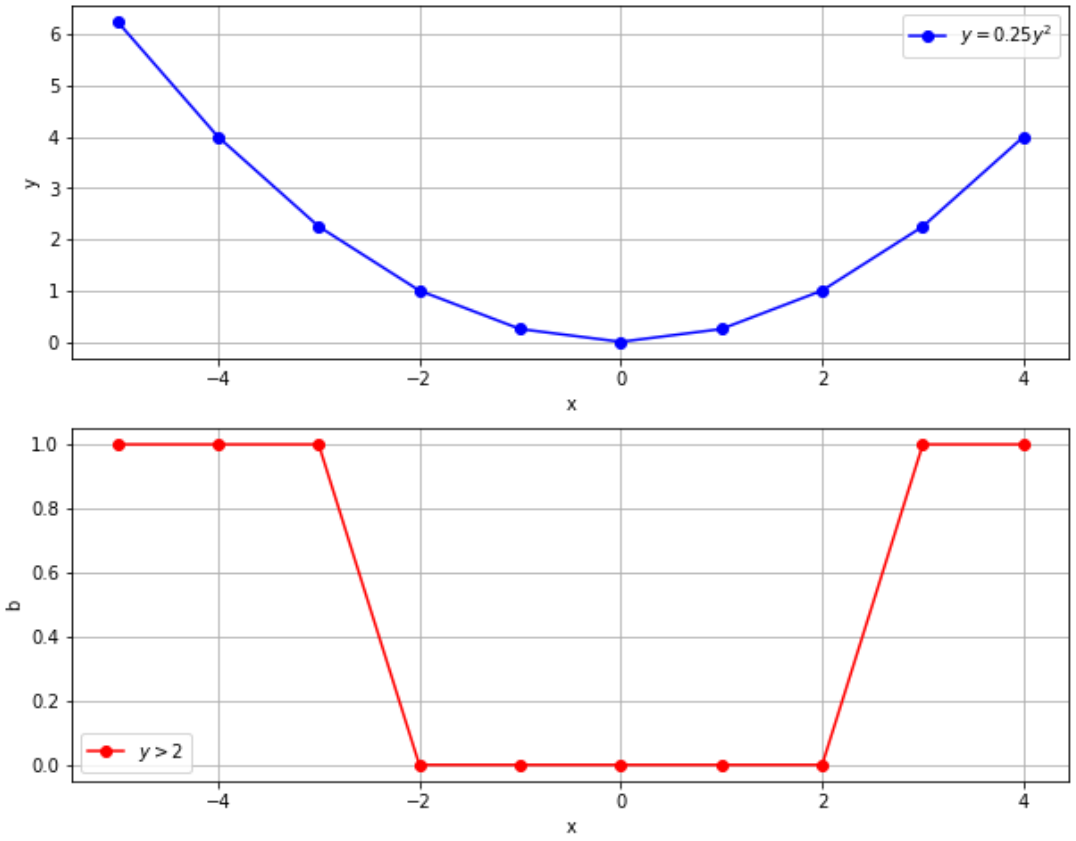
[15]:
```
array([0. , 0. , 0. , 1.5, 2. , 2.5, 0. , 0. , 0. , 0. ])
```

Graphing boolean values

[4]:
```
fig, ax = plt.subplots(2, figsize = (10, 8))

plt.sca(ax[0])
plt.plot(x, y, 'o-', color = 'b', label = '$y = 0.25y^2$')
plt.grid(True)
plt.xlabel('x')
plt.ylabel('y')
plt.legend()

plt.sca(ax[1])
plt.plot(x, b, 'o-', color = 'r', label = '$y > 2$')
plt.grid(True)
plt.xlabel('x')
plt.ylabel('b')
plt.legend()
plt.show()
```



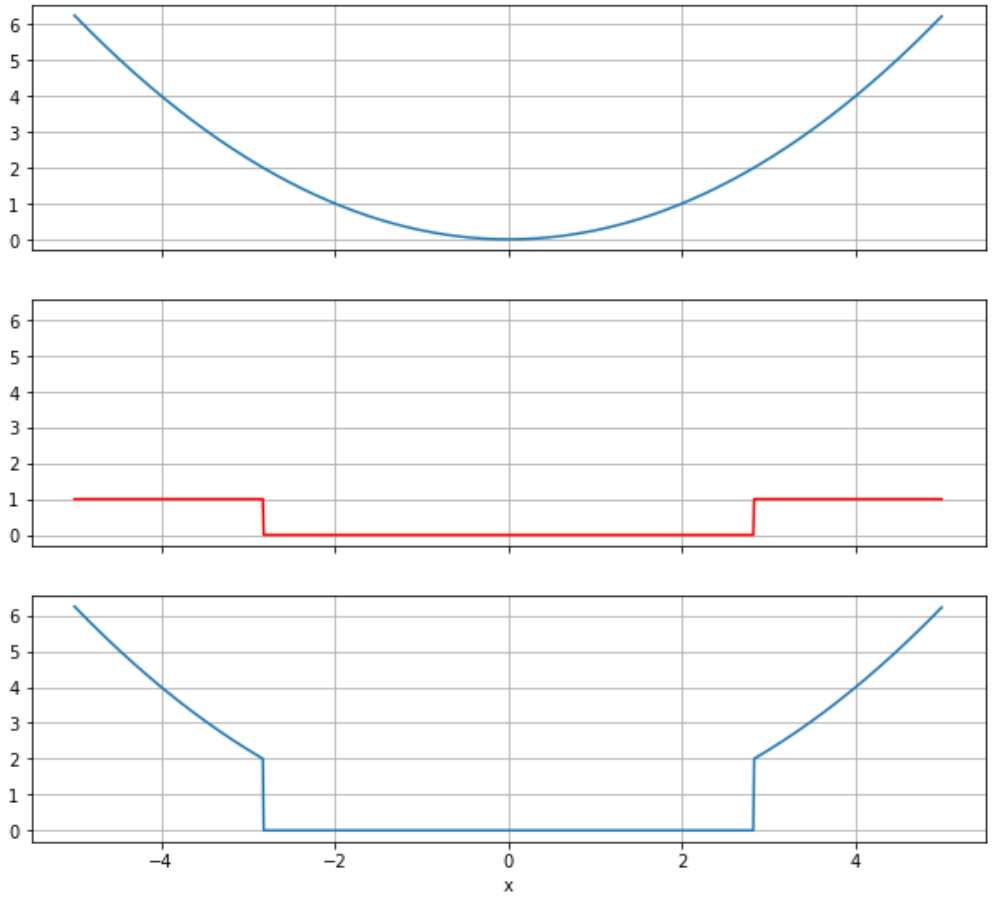
Example – multiplying numerical array with boolean values

```
[5]: x = np.arange(-5, 5, 0.01)
     y = 0.25*x**2
     b = y > 2

     fig1, ax = plt.subplots(3, figsize = (10, 9),
                             sharey = True, sharex = True)
     plt.sca(ax[0])
     plt.plot(x, y)
     plt.grid(True)

     plt.sca(ax[1])
     plt.plot(x, b, color = 'r')
     plt.grid(True)

     plt.sca(ax[2])
     plt.plot(x, b*y)
     plt.grid(True)
     plt.xlabel('x');
```



Example – piecewise continuous function

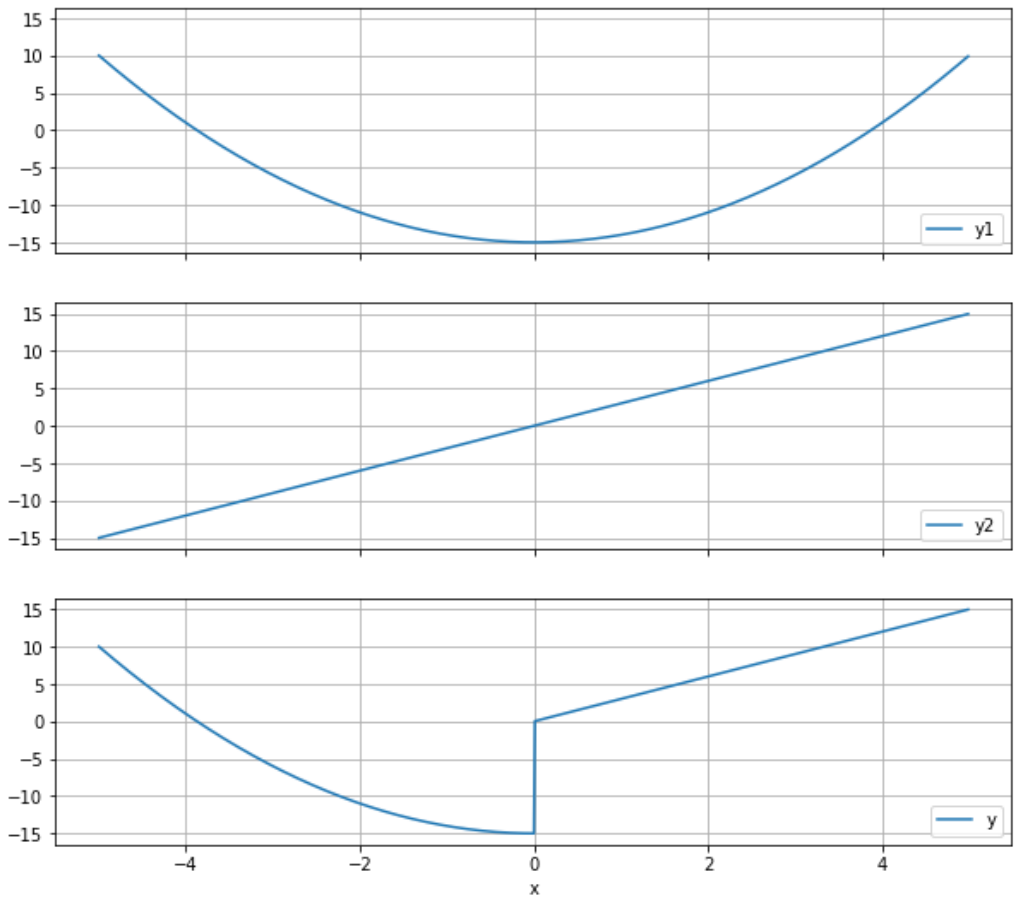
```
[6]: x = np.arange(-5, 5, 0.01)
     y1 = x**2 - 15
     y2 = 3*x
     y = (x < 0)*y1 + (x >= 0)*y2

     fig1, ax = plt.subplots(3, figsize = (10, 9),
                             sharey = True, sharex = True)

     plt.sca(ax[0])
     plt.plot(x, y1, label = 'y1')
     plt.grid(True)
     plt.legend(loc = 'lower right')

     plt.sca(ax[1])
     plt.plot(x, y2, label = 'y2')
     plt.grid(True)
     plt.legend(loc = 'lower right')

     plt.sca(ax[2])
     plt.plot(x, y, label = 'y')
     plt.grid(True)
     plt.legend(loc = 'lower right')
     plt.xlabel('x');
```



Next steps

- Practice – Lab 3
  - Notebook can be found from OMA assignments
  - Moodle has code checking and verify
- Read more
  - Numpy documentation
    - Indexing on ndarrays